Attracted by light: vision-based steering virtual characters among dark and light obstacles

Axel López, François Chaumette, Eric Marchand, Julien Pettré

Attracted by light: vision-based steering virtual characters among dark and light obstacles

Axel López
axel.lopez-gandia@irisa.fr
Univ Rennes, Inria, CNRS, IRISA, France

François Chaumette
francois.chaumette@inria.fr
Univ Rennes, Inria, CNRS, IRISA, France

Eric Marchand
eric.marchand@irisa.fr
Univ Rennes, Inria, CNRS, IRISA, France

Julien Pettré
julien.pettre@inria.fr
Univ Rennes, Inria, CNRS, IRISA, France

ABSTRACT

This paper introduces the use of numerical optical flow (OF) in vision-based steering techniques - that control characters locomotion trajectories by using a simulation of their visual perception. In contrast with synthetic OF that was previously used, numerical OF is sensitive to the contrast of objects, and provides, for example, uncertain results in dark areas. Thus, we here propose a locomotion control technique which is robust to such uncertainty: dark areas in the scene are processed as obstacles, that however may be traversed in case of necessity. As demonstrated in various scenarios, this tends to make character avoiding darkest areas, or traversing them more carefully, as it can be observed for real humans.

CCS CONCEPTS

• **Computing methodologies** → **Modeling and simulation**; **Model development and analysis**; *Model verification and validation*; *Animation.*

KEYWORDS

Animation, Optical Flow, Character navigation

ACM Reference Format:
Axel López, François Chaumette, Eric Marchand, and Julien Pettré. 2019. Attracted by light: vision-based steering virtual characters among dark and light obstacles. In *Motion, Interaction and Games (MIG '19), October 28–30, 2019, Newcastle upon Tyne, United Kingdom.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3359566.3360085

1 INTRODUCTION

The objective of a virtual human simulation is to reproduce the mechanisms by which humans behave in some situations. We are interested in human locomotion, and more specifically, their displacement based on the visual information about their environment. Many solutions have been proposed to allow a virtual human to find his way in large environments (motion planning) and to adjust his trajectory locally to the presence of obstacles (local method). It is this second problem that we consider in this paper.

The problem has been addressed in the purpose of creating autonomous agents or simulating crowds. Most solutions are based on a geometric formulation of the problem, solved by applying systems of rules or by defining energy function that guide the agent towards its goal while avoiding obstacles. More recent approaches simulate the visual perception of agents [Ondřej et al. 2010] [Dutra et al. 2017], and made finally possible to control the locomotion of an agent solely on the basis of information about the optical flow generated by its movement relative to the environment [López et al. 2019]. We base our current work on this previous result.

The idea behind [López et al. 2019] is to limit itself to using only visual information, the optical flow and its local characteristics, known to be directly perceptible by humans and used in controlling their movement [Warren et al. 2001]. However, in this previous work, a synthetic optical flow is calculated (querying the simulation). To further improve the realism of the principle by which virtual humans are simulated, this relative motion should be considered unknown and deduced by a successive perception of the environment over time.

This is the objective we set in this work. We replace here the calculation of a synthetic optical flow by a technique based on the use of successive images of the environment perceived from the agent's point of view. In doing so, we face the uncertainty in areas with low contrast and in particular dark areas, which generate unreliable optical flow. It is then necessary to take this uncertainty into account, by making our agents avoid them when possible. Doing so, we reproduce human behaviour whose locomotion trajectories are sensitive to the lighting conditions of a given environment. The contribution of this work is therefore to propose the first virtual human steering algorithm that implicitly takes into account the lighting conditions of the environment.

2 BACKGROUND

The problem we address in this paper covers related work in different fields, including Computer Graphics and Robotics.

2.1 Character steering

The goal of character steering is to enable a virtual agent to navigate in a virtual environment, e.g., to reach a certain destination while avoiding obstacles. Some of the first models proposed consider virtual agents as point particles. Then, the interaction of the agent with the rest of the world is modeled with attractive (goal) and repulsive (obstacles) forces. Social forces [Helbing and Molnár 1995] are a well-known example. Several improvements to this model have been proposed [Karamouzas et al. 2009] [Karamouzas et al. 2014] [Bouvier et al. 1997].

Seeking greater realism, Reynolds combined different small task to enable more complex behaviors, pioneering rule-based models

MIG '19, October 28–30, 2019, Newcastle upon Tyne, United Kingdom
https://doi.org/10.1145/3359566.3360085

[Reynolds 1987] [Reynolds 1999]. Following this, techniques were proposed to enable safe trajectories for virtual agents. RVO [van den Berg et al. 2008] [van den Berg et al. 2011] is an efficient collision avoidance approach looking for collision free velocities for agents in the near future, also explored in [Paris et al. 2007] and imropoved by several authors [Patil et al. 2011] [Wolinski et al. 2016].

2.2 Vision-based character steering

In order to improve realism of virtual characters, many approaches attempt to closely reproduce the human perceptual system. Tu et al. [Tu and Terzopoulos 1994] searched to reproduce the entire locomotion system of fishes and was improved later by Terzopoulos et al. [Terzopoulos and Rabie 1995]. Ondrej et al. [Ondřej et al. 2010] and Dutra et al. [Dutra et al. 2017] presented collision avoidance models using synthetic vision. Most of the previous methods perceive geometrical information that is then processed to steer agents. This is still far from the way a real human perceives its environment. In hopes to close this gap, López et al. [López et al. 2019] presented a steering model based on optical flow as it is known to play an important role in human and animal navigation.

2.3 Vision-based robot motion control

The robotics community has explored the use of vision to solve the navigation problem. Braillon et al. [Braillon et al. 2006] designed a robot using variations in optical flow to detect obstacles. Souhila and Karim [Souhila and Karim 2007] explored the use of expansion centers in optical flow to detect obstacles and the time to collision to them. Zingg et al. [Zingg et al. 2010] used optical flow to estimate depth information and correct the trajectory of robots. Our work differs from these approaches in that optical flow not only detects obstacles but also guides the best strategy to avoid them allowing our model to handle dynamic environments.

2.4 Positioning

Our approach extends the vision-based steering approaches. More specifically, it extends the approach presented in [López et al. 2019]. Our current work is based on the same principles for controlling the characters' locomotion: visual features extracted from its visual neighbor and directly used to control its self motion. These features are directly related to the optical flow (OF), i.e., the apparent motion of objects with respect to the character's perspective. In [López et al. 2019], a synthetic OF is computed (the relative motion of objects known) however, in nature, the perception of OF depends on the perceived variation of light. Therefore, contrast is required to allow registering the visual positions of objects in time and integrating this information to estimate OF. In this work, we thus explore OF computation based sequences of images of the scene. This results in inaccuracies in places of the scenes where objects lack contrast, e.g., in dark areas. Following sections describe how dark objects and areas are handled in the process of steering characters in the scene.

3 NAVIGATION MODEL

This section summarizes the proposed agent model to enable navigation in dynamic environments and changing lightning conditions. Our agents run a vision-based control loop illustrated in Figure 1 similar to the one presented in [López et al. 2019] to navigate in their environment.

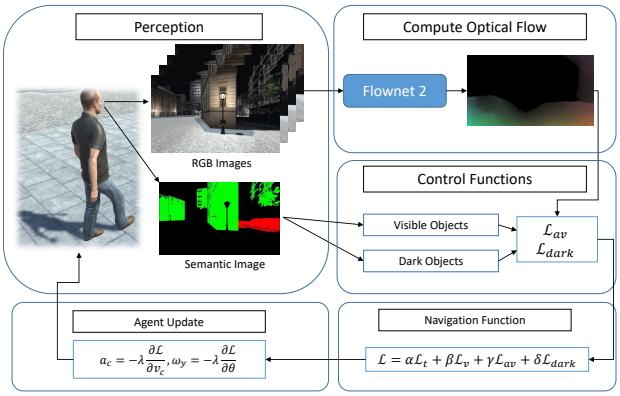

Figure 1: Control Loop summary. At each iteration, agents start at the perception stage to obtain a sequence of RGB and semantic data images. Optical flow is then computed from the RGB images. From the semantic information, obstacles are segmented. Finally, the agent state is updated by minimizing the navigation function.

3.1 Agent Perception

The highlight of our steering method is the use of optical flow, a quantity that is known to be used by humans and animals. In a similar way to humans, agents perceive the environment through virtual cameras placed in their head. At every iteration of the control loop, these cameras capture, RGB and semantic images. The RGB image is a render of the scene from the point of view of the agent. Two RGB images from consecutive frames are processed to compute optical flow using the FlowNet2.0 [Ilg et al. 2017] solver. We provide more details on optical flow in Section 4. The second image is rendered to encode semantic information so that every pixel has a label that allows our agent to know what kind of object lies behind and if there is a boundary between two objects. Pixels that are too dark in the RGB image are assigned a special label to identify them as dark pixels. Details on how to obtain the semantic information are explained in Section 5.

3.2 Visual Features

The second stage consists on extracting the relevant information from the images acquired. In order to do this, we first need to identify collections of pixels belonging to individual visible objects. We use the semantic image to segment different objects and remove the ground and sky from obstacles. For each visible object a set of visual features are collected and computed. First, the set of optical flow vectors $\mathbf{u}_j$ for each pixel i corresponding to that object. From the position of the pixels the center g_{xi} and size of the object is computed. Then, using the optical flow vectors we compute the Focus of Expansion f_{xi}, which enables us to detect future collision. Finally, we compute the time-to-collision τ_i of the object with the image plane. For dark objects only their position and size are computed.

3.3 Control Functions

Control functions define a very specific task such as goal reaching or collision avoidance. They are error functions that penalize undesired state of the agent. They use the visual features defined in the previous section to define their tasks. Most of the following, were introduced and demonstrated in [López et al. 2019], we extended them to consider differently light and dark objects.

3.3.1 Target Reaching and Speed Control. In order to reach a certain point in space, the agent needs to align its direction of motion to point towards this point. This is equivalent to move t_x to the center of the field of view. In addition, we want to define a preferred speed v_c^* for our agent. To this end the following functions are used,

$$\mathcal{L}_t = \frac{1}{2}t_x^2,\ \mathcal{L}_v = \frac{1}{2}(v_c - v_c^*)^2, \tag{1}$$

with v_c being the current velocity of the agent.

3.3.2 Collision avoidance. As said in Section 3.2 the FOE f_{xi} is used to prevent collisions. The condition for a future collision is that the FOE overlaps its corresponding object. To this end we use the following control function,

$$\mathcal{L}_{av} = \sum_i^{visible} I(\tau_i) \exp\left(-\frac{|g_{xi} - f_{xi}|}{\sigma_i}\right), \tag{2}$$

with $I(\tau_i)$ being a linear function that decreases the weight of an obstacle proportional to its time-to-collision τ_i and σ_i as measure of the object's width in the image.

This function iterates over all visible obstacles and penalizes the FOE and the center of the object from being too close. Obstacles with a lower time-to-collision induce a greater penalization in the control function.

Finally, we propose a very similar equation to avoid uncertain or dark obstacles,

$$\mathcal{L}_{dark} = \sum_i^{dark} \exp\left(-\frac{|g_{xi}|}{\sigma_i}\right). \tag{3}$$

This control function iterates over all uncertain objects, penalizing any motion towards these objects.

Navigation functions express a more complex task by combining multiple control functions. In this paper we use the following navigation function which allow agents to perform collision-free target reaching tasks in static and dynamic environments.

$$\mathcal{L} = \alpha\mathcal{L}_t + \beta\mathcal{L}_v + \gamma\mathcal{L}_{av} + \delta\mathcal{L}_{dark}, \tag{4}$$

with $\gamma \geq \delta$ so that visible objects have higher priority over dark objects.

3.4 Agent Update

The control variables of the agent are acceleration a_c and angular velocity ω_{cy}. They are updated by performing a gradient descent on $\mathcal{L}$, therefore,

$$a_c = -\lambda\frac{\partial\mathcal{L}}{\partial v_c},\ \omega_{cy} = -\lambda\frac{\partial\mathcal{L}}{\partial\theta}, \tag{5}$$

with λ being the step size parameter.

In the following sections we will further explain contributions of this paper. First, we describe optical flow and how a numerical solution can be used in place of the ideal equations (cf. Section 4). Second, we explain how to identify visible and dark objects in the field of view (cf. Section 5). Finally, we describe our approach to make the agent aware of these dark regions in order to avoid them (cf. Section 6).

4 OPTICAL FLOW

The key element in our steering model is using optical flow to drive navigation. In this section we recall the definition of optical flow and how it is computed.

4.1 Definition of Optical Flow

Optical flow is defined as the apparent motion of objects and it is produced by the variation of colors from frame to frame. Like in the model presented by López et al. [López et al. 2019], by considering the approximation that the flow is the projected motion of object in the scene, and removing the rotational components, optical flow is written as,

$$\begin{cases} u_j = \left(v_{ix} - x_j v_{iz}\right)/Z_j, \\ v_j = \left(v_{iy} - y_j v_{iz}\right)/Z_j, \end{cases} \tag{6}$$

with $(x_j, y_j) = (X_j/Z_j, Y_j/Z_j)$ being the projection of a 3D point $\mathbf{X}_j$ in the image plane.

4.2 Computing Optical Flow

In [López et al. 2019], a synthetic optical flow was generated reproducing perfectly Equation (6). However, humans can only perceive optical flow from the variation of perceived images. In order to replicate this, we compute optical flow numerically from two consecutive frames. This way of computing optical flow consists on finding correspondences between pixel from one frame to the next one. We use FlowNet2.0 [Ilg et al. 2017], a deep learning approach, as this solution is a good compromise between accuracy and performance. In order to effectively remove the rotational flow, the rotation of the agent is temporarily stopped so optical flow. Even though, this method of computing optical flow does not solve Equation (6), we claim and demonstrate that it provides a good enough approximation for the optical flow from the numerical solver. Figure 2 shows an example of numerical optical flow. In the accompanying video we show additional comparisons between numerical and synthetic optical flow. Despite the noise inherent to numerical flow, we demonstrate that it can still be used for character navigation while being more realistic.

4.3 Parameters

Numerical optical flow, as opposed to synthetic flow, is sensitive to various parameters of the simulation. We used the unity engine with its standard shader to lit the scenes. In dark scenarios ambient light is reduced. We use the FlowNet2.0 network to compute numerical optical flow from two consecutive images with a delay of 33ms between them. To solve aliasing artifacts we render images with a resolution of 1024x1024 which are then down sampled to 512x512.

5 VISIBLE AND DARK OBJECTS

The visual perception system of our agents is capable of recording semantic information. In order to do this, every object in the scene

is assigned an id value. Then, the scene is rendered but instead of color, every pixel encodes this id. In a second scan of the image, we check the RGB image to detect dark pixels. When a pixel is detected as dark its id is replaced by an special id common for every dark pixels. This allows our agents to extract two types of objects, visible and dark. Visible objects are segmented from one another using the semantic image and their visual features are computed (FOE, time-to-collision, etc). Visible pixels belonging to the ground are discarded as the ground should not be considered an obstacle. Pixels belonging to the sky are always discarded regardless of their label. Dark objects objects often lack contrast making the optical flow solver unreliable for these pixels. As a result we cannot compute visual features related to optical flow and only know the location and size of these regions. In addition, we discard dark objects which are in the upper half of the image as there is no collision risk with objects above the ground.

Figure 2 illustrates a semantic image in a given situation. The objects at the left, tree and buildings are marked as visible objects in green. Dark pixels are marked in red such as the shadow region on the right of the scene. Pixels in black are contours or discarded pixels belonging to visible ground or sky. We also discard any object with a low number of pixels relative to the size of the image.

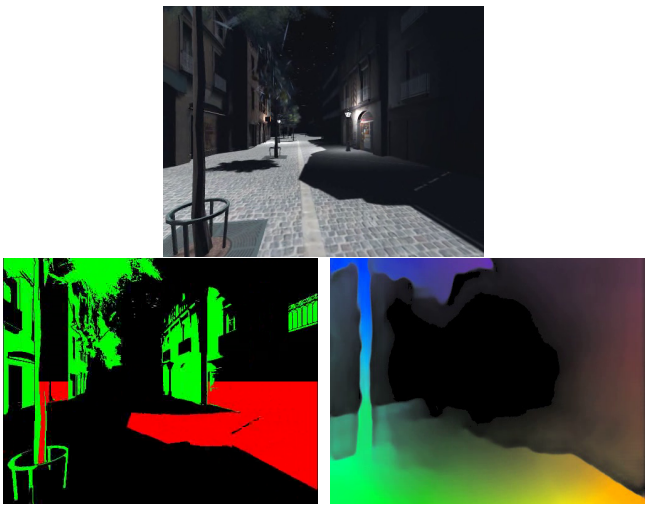

Figure 2: Semantic information. (top) Color image of the scene. (bottom left) Label image, green regions indicate visible obstacles, black regions are discarded regions, red regions are pixels labelled as uncertain. (bottom right) Numerical optical flow.

6 AVOIDING DARK OBJECTS

After image segmentation, two kinds of objects exists, visible objects and dark objects as explained in the previous section. Visible objects are processed in Equation (2) using visual features extracted from optical flow to avoid these obstacles as it reaches the goal.

As seen in Figure 2 optical flow solver produces inaccurate results in dark regions due to the lack of color and contrast. This renders optical flow computed for dark regions unreliable. Therefore, we consider that dark regions may contain obstacles and should be avoided. As we only know the position and size of these objects they are processed in Equation (3) as a dark region avoidance function. Then in the navigation function in Equation (4) the contribution of $\mathcal{L}_{dark}$ is equal or lower than $\mathcal{L}_{av}$. The reason for this is that any visible obstacle should have a greater weight than uncertain ones.

7 RESULTS

In this section we test the control scheme presented in Section 3 in several scenarios. We demonstrate that our control loop enables our agents to navigate dynamic environments in various lightning conditions.

7.1 Control Variables

In the following examples we use Equation (4) to navigate our agents toward the goal while avoiding obstacles. The gradient of this function is used to update the control variables of the agent at every iteration of the control loop. They have the following expression,

$$
\begin{aligned}
a_c = -\lambda \Bigg[& \beta(v_c - v_c^*) + \\
& \gamma \sum_i^{visible} I(\tau_i) \exp\left(-\frac{|g_{xi} - f_{xi}|}{\sigma_i}\right) \frac{sign(\Delta x_i)}{\sigma_i} \frac{-f_{xi}\tau_i}{Z_i} \Bigg], \\
\omega_{cy} = -\lambda \Bigg[& -\alpha t_x (t_x^2 + 1) + \\
& \gamma \sum_i^{visible} I(\tau_i) \exp\left(-\frac{|g_{xi} - f_{xi}|}{\sigma_i}\right) \frac{sign(\Delta x_i)}{\sigma_i} \left(f_{xi}^2 - g_{xi}^2 - \frac{\tau_i}{Z_i} v_c\right) \\
& + \delta \sum_i^{dark} \exp\left(-\frac{|g_{xi}|}{\sigma_i}\right) \frac{sign(g_{xi})}{\sigma_i} \left(-(g_{xi}^2 + 1)\right) \Bigg],
\end{aligned}
\tag{7}
$$

with $\alpha = \beta = \gamma = \delta = 1$ and $v_c^* = 1m/s$ for all experiments

In this work we do not use any knowledge about the depth Z_i of the objects. In order to compensate this missing information we make the following approximation $\frac{\tau_i}{Z_i} v_c = \frac{1}{2}$, which is true when every object is moving at the same as the agent. This approximation allows our agent to avoid obstacles as long as the velocity of other objects is not very different from the agent's and is still valid for static obstacles.

7.2 Numerical VS Synthetic Optical Flow

We present a few examples to compare the results of using numerically computed optical flow with synthetic flow used by [López et al. 2019]. Figure 3 shows an agent navigating in a street scenario with static obstacles. The results are very similar in both situations as for static obstacles the numerical optical flow is capable of giving a good estimation of the flow.

The accompanying video provides additional comparisons showing the differences of using numerical and synthetic optical flow.

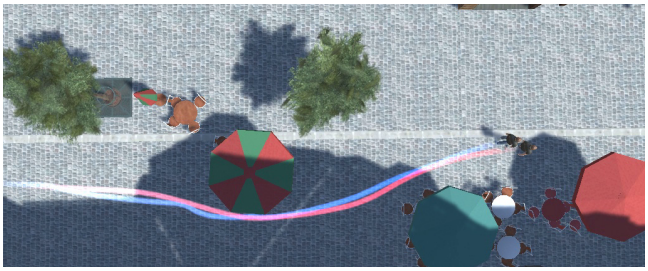

Figure 3: Comparison of trajectories with different optical flow computation methods in a street with obstacles. (red) Synthetic optical flow and (blue) numerical optical flow.

7.3 Static Lightning

In this section we show how our algorithm perform when agents are tasked to navigate in environments with dark and shadowed areas.

Figure 4 shows an agent moving in a street with a very hard shadow at his right. We compare the resulting trajectory when taking into account the light conditions (blue trajectory) and without any light perception (red agent). The blue agent maintains a certain distance with the shadow when possible. However it decides to step a bit into it to maintain a certain distance with the tree on his left. Shadows and uncertain objects take lower priority than visible obstacles. The red agent just walks normally keeping distance between him and the trees ignoring the shadow completely.

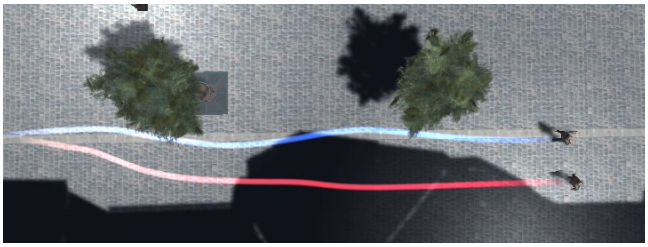

Figure 4: Agent trajectory in dark street. The blue trajectory is an agent using our navigation model to avoid dark regions, the red trajectory is an agent using the model presented in [López et al. 2019].

Figure 5 shows an agent in a street with two possible ways, one is lit and the other is dark. The dark street is detected as a dark obstacle by the blue agent. Therefore, the agent moves to avoid it and in turn takes the path with greater visibility. The red agent does not take into account the shadow and as the dark street is closer, prefers this choice over the lit path.

Additional examples are provided in the accompanying video.

7.4 Dynamic Lightning

In this section we present how our algorithm can compensate for variations in light conditions.

Figure 6 shows an agent navigating a very dark environment. It carries a light source in a dark scene. The agent reacts to obstacles only as soon as they are perceived as visible obstacles. This causes a late reaction to avoid obstacles.

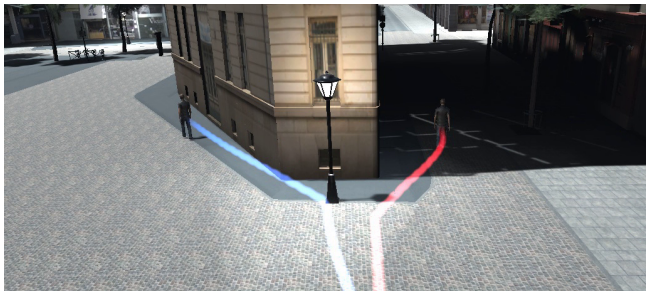

Figure 5: Lit and dark street. The path of the agent forks in two streets. The blue trajectory is an agent using our navigation model to avoid dark regions, the red trajectory is an agent using the model presented in [López et al. 2019].

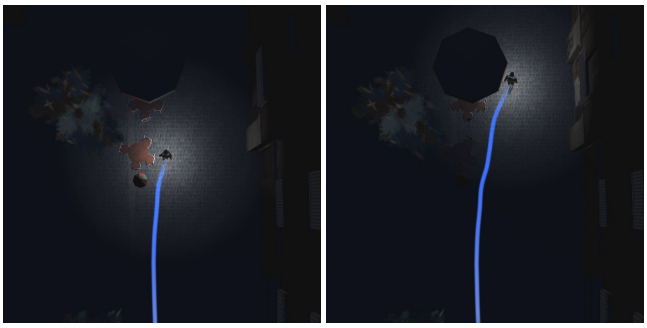

Figure 6: Agent trajectory in a dark street. The agent carries a light source which is the only light source in the scene.

7.5 Performance

In this section we give indicative numbers about our approach performance. Simulations run on a 3.19GHz Intel Xeon processor, 32GB of RAM, a GTX 1080 Ti and the Unity engine to render the scene.

The performance of our algorithm depends on the resolution used by the agents to perceive the environment. With a resolution of 256^2 every iteration of the control loop takes 100ms, for 512^2 it takes 200ms and for 1024^2 it is 750ms. RGB and semantic images are downloaded from th GPU to the CPU which may be very time-consuming. For this reason, we believe that a full GPU implementation of our approach would result in a big performance boost. Finally, optical flow is a time-consuming operation as the FlowNet2.0 network is composed of several convolution layers.

8 DISCUSSION

In the previous section we have demonstrated our steering method in various scenarios: static and dynamic environments, made of dark and lit areas and objects. The main difference with the related previous approach [López et al. 2019] is the use of numerical optical flow. This method generates uncertainty in the estimation of

the optical flow which is due to lack of contrast. Our agents are programmed to avoid these areas to prevent possible collisions.

8.1 Realism

The goal of this work is to simulate human behavior by reproducing their perceptual system. Our agents have only access to RGB images and semantic information similar to humans. Optical flow encodes sufficient information about the relative motion of obstacles to allow agents to identify risk of collision and avoid it. The advantage of our method is that it does not require prior knowledge of the scene other than semantic information.

Light conditions change the way humans navigate an environment as they need to account for lack of information in their field of view. We have demonstrated that our model takes into account these conditions and changes the behavior of the agent accordingly.

Our method could be used to further explore how real humans perceive their environment and the navigation strategies they perform in different light conditions. However, this would require acquisition of real human data in a controlled environment, which is out of the scope of this paper.

8.2 Limitations and Future Work

As our goal is to closely reproduce human perception, the amount of information our agents perceive is at the same time diminished and expanded with respect to other geometrical approaches. It is expanded as we are now able to react to light conditions, which is not supported by other methods. But we loose precise knowledge of the motion of objects in the scene, which may introduce inaccuracies and higher risks of collisions. In addition, optical flow solvers are far from perfect and introduce a high amount of noise that may produce wrong results. Also, we still require prior knowledge of semantic information. Humans are very efficient at identifying different obstacles in their field of view. This is an open problem in the computer vision community but we believe that a deep learning approach [Zhao et al. 2017] could supply this information. As we use a local optimization approach, an agent may still face the issue of local minimum.

9 CONCLUSIONS

We presented a new steering model based on synthetic vision to allow agents to react to changing light conditions. We described a method to identify regions with no information in the field of view of the agent and showed how to process them to allow agents to follow a safer trajectory. We have shown our method in several scenarios. Our approach moves a step closer to allow robots to navigate real environments as we rely only on RGB images and semantic information. RGB images are trivial for a robot to obtain and although it is an open problem, semantic information from images has received a lot of attention from the computer vision community.

REFERENCES

E. Bouvier, E. Cohen, and L. Najman. 1997. From crowd simulation to airbag deployment: particle systems, a new paradigm of simulation. *J. of Elec. Imag.* 6 (1997), 6 – 14.

C. Braillon, C. Pradalier, J. L. Crowley, and C. Laugier. 2006. Real-time moving obstacle detection using optical flow models. In *IEEE Intelligent Vehicles Symposium*. 466–471.

T. B. Dutra, R. Marques, J.B. Cavalcante-Neto, C. A. Vidal, and J. Pettré. 2017. Gradient-based steering for vision-based crowd simulation algorithms. *Computer Graphics Forum* 36, 2 (2017), 337–348.

D. Helbing and P. Molnár. 1995. Social force model for pedestrian dynamics. *Phys. Rev. E* 51 (May 1995), 4282–4286. Issue 5.

E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox". 2017. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

I. Karamouzas, P. Heil, P. van Beek, and M. H. Overmars. 2009. A Predictive Collision Avoidance Model for Pedestrian Simulation. In *Motion in Games*, Arjan Egges, Roland Geraerts, and Mark Overmars (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 41–52.

I. Karamouzas, B. Skinner, and S. J. Guy. 2014. Universal Power Law Governing Pedestrian Interactions. *Phys. Rev. Lett.* 113 (Dec 2014), 238701. Issue 23.

A. López, F. Chaumette, E. Marchand, and J. Pettré. 2019. Character navigation in dynamic environments based on optical flow. *Computer Graphics Forum* 38, 2 (2019), 181–192.

J. Ondřej, J. Pettré, A. Olivier, and S. Donikian. 2010. A Synthetic-vision Based Steering Approach for Crowd Simulation. *ACM Trans. Graph.* 29, 4, Article 123 (July 2010), 123:1–123:9 pages.

S. Paris, J. Pettré, and S. Donikian. 2007. Pedestrian Reactive Navigation for Crowd Simulation: a Predictive Approach. *Computer Graphics Forum* 26, 3 (2007), 665–674.

S. Patil, J. van den Berg, S. Curtis, M. C. Lin, and D. Manocha. 2011. Directing Crowd Simulations Using Navigation Fields. *IEEE Trans. on Visualization and Computer Graphics* 17, 2 (Feb 2011), 244–254.

C. Reynolds. 1987. Flocks, Herds and Schools: A Distributed Behavioral Model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*. ACM, New York, USA, 25–34.

C. Reynolds. 1999. Steering Behaviors For Autonomous Characters. *Game Developers Conference* (1999), 763–782.

K. Souhila and A. Karim. 2007. Optical Flow Based Robot Obstacle Avoidance. *International Journal of Advanced Robotic Systems* 4, 1 (2007), 2.

D. Terzopoulos and T. F. Rabie. 1995. Animat vision: Active vision in artificial animals. In *Proceedings of IEEE International Conference on Computer Vision*. 801–808.

X. Tu and D. Terzopoulos. 1994. Artificial Fishes: Physics, Locomotion, Perception, Behavior. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '94)*. ACM, New York, USA, 43–50.

J. van den Berg, Stephen J. Guy, Ming Lin, and Dinesh Manocha. 2011. Reciprocal n-Body Collision Avoidance. In *Robotics Research*, Cédric Pradalier, Roland Siegwart, and Gerhard Hirzinger (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 3–19.

J. van den Berg, M. Lin, and D. Manocha. 2008. Reciprocal Velocity Obstacles for real-time multi-agent navigation. In *IEEE Int. Conf. on Robotics and Automation*. 1928–1935.

WH Warren, BA Kay, WD Zosh, AP Duchon, and S Sahuc. 2001. Optic flow is used to control human walking. *Nature neuroscience* 4, 2 (February 2001), 213—216.

D. Wolinski, M. C Lin, and J. Pettré. 2016. WarpDriver: context-aware probabilistic motion prediction for crowd simulation. *ACM Transactions on Graphics* 35, 6 (2016).

H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia. 2017. ICNet for Real-Time Semantic Segmentation on High-Resolution Images. *arXiv preprint arXiv:1704.08545* (2017).

S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart. 2010. MAV navigation through indoor corridors using optical flow. In *2010 IEEE International Conference on Robotics and Automation*. 3361–3368.